

PATENT
POU920010126US1IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)
GIAMPAOLO LAURIA, et al.) Group Art Unit: 2177
Serial No.: 10/092,321)
Filed: March 6, 2002) Examiner: K. Lu
For: AUTOMATIC FILE SYSTEM)
MAINTENANCE)

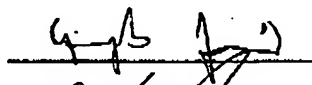
37 C.F.R. § 1.131 DECLARATION

We, the undersigned, are the Applicants for the above-identified patent application and hereby declare the following:

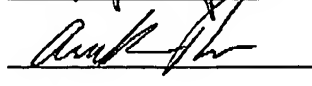
- 1) The pending claims of our above-identified patent application were rejected under 35 U.S.C. § 102(e) and 35 U.S.C. § 103(a) based on U.S. Patent Application Publication No. US 2003/0110190 to Achiwa et al., which is entitled "Method and System for File Space Management" and published on June 12, 2003 ("Achiwa"). The Achiwa reference has a 35 U.S.C. § 102(e) date of December 10, 2001.
- 2) The invention claimed in the above-identified patent application was reduced to writing in the United States prior to the December 10, 2001 date of the Achiwa reference. Attached hereto is the relevant portion of an Invention Disclosure on which the above-identified patent application was based. This Invention Disclosure was prepared prior to December 10, 2001.

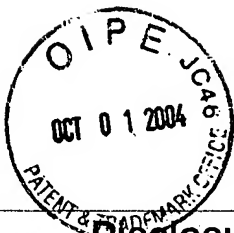
We, the undersigned, hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Name: Giampaolo Lauria

Signature: Date: 9-23-2004

Name: Anthony F. Pioli

Signature: Date: 9/23/04



1 KG



Disclosure POU8-2001-0091

Prepared for and/or by an IBM Attorney - IBM Confidential

Created By: Giampaolo Lauria Created On: [REDACTED]

Last Modified By: Stephanie C Wilson Last Modified On: [REDACTED]

Required fields are marked with the asterisk (*) and must be filled in to complete the form.

*Title of disclosure (in English)

FSDiet

Summary

Status	Final Decision (File)
Docket Family	POU9-2001-0126
Processing Location	POU
Functional Area	3Q - [REDACTED]
Attorney/Patent Professional	[REDACTED] Poughkeepsie/IBM
IDT Team	[REDACTED]
Submitted Date	[REDACTED]
Owning Division	SD
Incentive Program	
Lab	
Technology Code	
PVT Score	No PVT score has been calculated. To calculate a PVT score, press the 'Calculate' button.

Inventors with a Blue Pages entry

Inventors: Giampaolo Lauria/Poughkeepsie/IBM, Anthony Pioli/Poughkeepsie/IBM

Inventor Name	Inventor Serial	Div/Dept	Inventor Phone	Manager Name
> Lauria, Giampaolo	[REDACTED]		[REDACTED]	Oliva, Ed
Pioli, Anthony F.	[REDACTED]		[REDACTED]	Merritt, Laura J.

> denotes primary contact

Inventors without a Blue Pages entry

IDT Selection

Select Functional Area

IDT Team: [REDACTED] oughkeepsie/IBM [REDACTED] oughkeepsie/IBM [REDACTED] oughkeepsie/IBM [REDACTED] oughkeepsie/IBM [REDACTED] oughkeepsie/IBM [REDACTED] oughkeepsie/IBM [REDACTED] oughkeepsie/IBM [REDACTED] oughkeepsie/IBM [REDACTED] oughkeepsie/IBM [REDACTED] oughkeepsie/IBM [REDACTED] oughkeepsie/IBM	Attorney/Patent Professional: [REDACTED]
--	--

Response Due to IP&L : [REDACTED]

***Main Idea**

1. Describe your invention, stating the problem solved (if appropriate), and indicating the advantages of using the invention.

FSDiet is a program used to maintain file systems and/or directories at a certain level of usage. The usage level may be measured in different ways. Examples are by space remaining, space used, files used, or files remaining. FSDiet will generate a list of files to be removed in order to maintain a target usage level. FSDiet can optionally perform the removal in addition to generating the file list.

2. How does the invention solve the problem or achieve an advantage, (a description of "the invention", including figures inline as appropriate)?

We will try to illustrate the advantages of FSDiet by way of an example. This is an example only and in no way should be taken as the only application of FSDiet.

Administrators are usually concerned with maintaining systems at a certain level of usage, e.g., "maintain 20% free space on /tmp". To solve this particular task, the administrator typically would look at the files in /tmp and locate the oldest files (or perhaps the largest files) contained in that directory. Once they had located these files, they would remove enough of them so that /tmp was at least 20% free. The process of determining which files to remove and then how many of them to actually remove would generally be done by the administrator by hand.

Administrators do however need to know when free space becomes an issue. When all the space is used and a user's file operation fails, the user would usually alert the administrator about the space problem. Another common way for an administrator to become aware of this space problem is by other monitoring systems that watch space usage to notify them. Since file systems are in constant use, these problems may arise at any time. In any case, when space must be made, the task of determining the set of files to remove must be done. FSDiet does not address the "notification" problem, but is intended as the tool to use to solve the underlying space problem - it can be run manually to make space "now", or it can at some fixed interval and therefore maintain a desired amount of free space. In the later case, the intent is that no insufficient space problems would arise in the first place.

The advantage that FSDiet gives is that it can perform the freeing task automatically without any interactions by the administrator. Current tools used by the administrator can help them generate a list of the oldest files (for example), but this not optimal information when trying to determine the files to remove since the size of the files determines how many of them need to be removed. FSDiet removes all these considerations and allows the administrator to have FSDiet automatically "maintain 20% free space on /tmp".

Below is the information FSDiet collects from the user before performing its work. The information is presented as sequence, but the actual collection may be done via command-line arguments, GUI elements, or possibly a sequence of questions.

1) The target of the reduction:

This is a full path name and can be either a directory or a file system. FSDiet will determine what the target type is automatically. Some of the options described below can only apply to one or the other target types. These restrictions are listed in each section below as needed.

2) Reduction type:

When picking the files to be removed, by default we look at their time stamp with the oldest files the first to be removed. Other methods are offered, and these are key features of FSDiet utility.

These alternative reduction methods are:

a) Reduce file system by keeping its size below a % number

This is the example used above - it allows the user to maintain a certain percentage of free space in a file system. Note that this is not a valid reduction type if the target type is a directory since directories generally have no predetermined capacity. An example of this action would be "remove files in /tmp so that it is 20% free."

b) Reduce target by removing a percentage of existing files

This option allows the removal of a percentage of the current number of files in the target. This can be applied to either file system or directory targets. An example of this action would be "remove 30% of the files in /home"

c) Reduce target by some fixed amount

This would reduce the target by some fixed size. An example of this action would be "reduce /tmp by 100 megabytes." If the requested reduction is more than the current size of the target, an error is reported. This can be applied to either file system or directory targets.

d) Reduce target so that its file usage is below a certain size.

This option would reduce the target to a specific size (or possibly smaller - it is possible that the last file deleted will drop the final size below the target size). The actual resulting target size would be no larger than the specified size. This can be applied to either file system or directory targets.

3) Use of Recursion:

Recursion in this case refers to how FSDiet treats sub-directories of the target. File systems can have directories mounted on them (the usual case) and directories can contain sub-directories, so this is a natural expression of recursion. This option determines if the operations that FSDiet is to perform include the sub-components of the target. The choices are simply to act recursively - include files in the subdirectories as part of the list of files to be examined, or to act non-recursively - **do not** include files in the subdirectories as part of the list of files to be examined.

4) Empty Files:

The output of FSDiet is a list of files to be removed, in order to satisfy the reduction requested. (FSDiet can also perform the actual file removals). As this list is being developed, empty files may be encountered. This option answers the question as to how to handle these empty files - should they be automatically added to the list since they are empty or should they only be added to the list if they meet the normal file selection criteria? For example, a recent, though empty, file may not be removed since older files exist and hence would be removed first.

5) Empty directories:

This is very similar to the Empty Files option described above but instead of empty files, this option deals with empty directories. Since users may wish to preserve the directory structure of the target, FSDiet provides this option so that empty directories are not removed.

6) Sort order:

Conceptually, once the list of file removal candidates is created, FSDiet will start at the "top" and remove files until the target reduction is achieved. (Assuming this was the action the user chose.) This option specifies what order the list is arranged in. By default, the oldest files are at the "top" and so the result would be that the oldest files would be deleted first. However, the user may wish to delete the largest files first, or perhaps even the smallest files first.

These are the current sort options we envision:

1) By date, oldest files first (this is the default sort order).

2) By size, largest files first.

3) User defined. Here we would allow the user to provide a command to "compare" two files. When FSDiet needs to compare two files to decide which comes first in the list, this command will be used to make that determination. The command will be called and passed two parameters - namely, the names of the files to compare (We may optionally pass more parameters for future use.) This command should return a code that indicates if the first file should come before or after the second file. It is also possible that no preference is given to either file by the user supplied command. In these cases, FSDiet will then choose the relative order of the two files.

Note that above we describe sort options, but an overarching "reverse" can be applied to the sort order as well. For example, if the user chooses to sort by size and chooses to reverse the sort, they will then have the largest files **last**, or said another way, the smallest files first.

7) Action taken:

As stated above, the output of FSDiet is a list of files to remove in order to satisfy the reduction requested. FSDiet provides this option to allow automatic actions to be taken on each file in the removal list. Thus, FSDiet can provide automated target sizes as discussed in the introduction. Currently we envision these actions:

1) Print file name to screen (the default action)

2) Delete the file

3) Run a user defined command on each file

8) File Inclusions and Exclusions:

This option provides the user with the ability to exclude (or include) certain files from the removal candidate list. Supported methods of exclusion/inclusion currently are these:

1) File name matching against a user-supplied regular expression

2) User ID of file owner

3) Group ID of file owner

These "filters" are used when FSDiet is selecting which files to consider for removal. If the file matches one of these criteria, it is either excluded completely from the removal list (for the exclusion case) or "passed on" in the inclusion case. (By "pass on", we mean that the file will make it to the next stage of selection which is based on the other criteria the user chose.)

9) File System Traversal:

As FSDiet traverses the target and its sub-directories (if any), it is possible that different file systems actually contain the files. This option specifies whether or not FSDiet should cross over into a file system other than that of the original target. The available choices are simply to allow traversal and to disallow traversal.

10) Continuous Mode:

FSDiet is usually run as needed. It can also run in a continuous mode unattended. The default behavior is to run once, but this option gives the user the choice of running FSDiet repeatedly as a daemon process.

3. If the same advantage or problem has been identified by others (inside/outside IBM), how have those others solved it and does your solution differ and why is it better?

We are not aware of other systems that perform the same function as FSDiet.

Related tools are "ls", "find", and "tmpwatch".

"find" is a tool used to locate files in a file system. This tool can locate files and sort them by age. This is a typical usage for this tool, and in fact that is how it would be used in the example above to find the list of oldest files. However, "find" cannot find a set of files of a total size.

"ls" is a tool to list files in a directory. It too can generate lists of files, but has the limitation that it cannot generate a certain sized set of files.

"tmpwatch" is a utility for removing files based on when they were last accessed. Again, this does not address the amount of space needing to be freed as FSDiet does.

4. If the invention is implemented in a product or prototype, include technical details, purpose, disclosure details to others and the date of that implementation.

FSDiet has been implemented as a minimal prototype in Perl. This work was done in the last week of [REDACTED] No disclosures of any kind have been made.

***Critical Questions (Questions 1-9 must be answered in English)**

***Question 1**

On what date was the invention workable? [REDACTED] Please format the date as MM/DD/YYYY (Workable means i.e. when you know that your design will solve the problem)

***Question 2**

Is there any planned or actual publication or disclosure of your invention to anyone outside IBM?

☐ Yes
☒ No

If yes, Enter the name of each publication or patent and the date published below.

Publication/Patent:

Date Published or Issued:

Are you aware of any publications, products or patents that relate to this invention?

☐ Yes
☒ No

If yes, Enter the name of each publication or patent and the date published below.

Publication/Patent:

Date Published or Issued:

***Question 3**

Has the subject matter of the invention or a product incorporating the invention been sold, used internally in manufacturing, announced for sale, or included in a proposal?

☐ Yes
☒ No